

CLAIMS

What is claimed is:

- 1 1. A method of designing a plurality of integrated circuits (ICs), said method
2 comprising:
3 partitioning a technology independent RTL (register transfer level) netlist
4 between said plurality of ICs.
- 1 2. A method as in claim 1 further comprising:
2 compiling a hardware description language (HDL) code, wherein said
3 technology independent RTL netlist is produced after compiling said
4 HDL code.
- 1 3. A method as in claim 2 wherein said ICs each comprise a programmable logic
2 device and wherein said partitioning comprises assigning a portion of said technology
3 independent RTL netlist to one of said plurality of ICs.
- 1 4. A method as in claim 2 further comprising:
2 mapping said technology independent RTL netlist to a selected technology
3 architecture.
- 1 5. A method as in claim 4 wherein said mapping is performed after said
2 partitioning.

- 1 6. A method as in claim 4 further comprising:
2 performing a place and route operation after said mapping to implement said
3 ICs in said selected technology architecture.

- 1 7. A method as in claim 4 further comprising:
2 optimizing a design of each of said ICs after said partitioning.

- 1 8. A method as in claim 7 wherein said optimizing optimizes each of said ICs by
2 removing duplicative logic or input/outputs.

- 1 9. A method as in claim 4 wherein said HDL code is created without regard to
2 said partitioning.

- 1 10. A method as in claim 7 wherein said optimizing and said mapping are
2 performed after said partitioning.

- 1 11. A method as in claim 4 further comprising:
2 mapping portions of said technology independent RTL netlist to a selected
3 technology architecture wherein estimates of IC resources are obtained
4 from said mapping portions and wherein said mapping portions is
5 performed after said compiling and before said mapping.

- 1 12. A method as in claim 4 further comprising:
2 optimizing interconnects between modules of said technology independent
3 RTL netlist before said partitioning.
- 1 13. A method as in claim 11 wherein said estimates are used to decide how to
2 perform said partitioning.
- 1 14. A method as in claim 13 wherein a user considers said estimates and selects a
2 command to decide how to perform said partitioning.
- 1 15. A method as in claim 4 wherein said ICs each comprise a programmable logic
2 device and wherein said method further comprises:
3 testing a prototype of a system with said ICs;
4 performing a synthesis of said HDL code to generate at least one Application
5 Specific Integrated Circuit (ASIC).
- 1 16. A machine readable medium containing a plurality of executable instructions,
2 which when executed on a digital processing system cause said digital processing
3 system to perform a method of designing a plurality of integrated circuits (ICs), said
4 method comprising:
5 partitioning a technology independent RTL (register transfer level) netlist
6 between said plurality of ICs.

1 17. A machine readable medium as in claim 16, wherein said method further
2 comprises:
3 compiling a hardware description language (HDL) code, wherein said
4 technology independent RTL netlist is produced after compiling said
5 HDL code.

1 18. A machine readable medium as in claim 17 wherein said ICs each comprise a
2 programmable logic device.

1 19. A machine readable medium as in claim 17, wherein said method further
2 comprises:
3 mapping said technology independent RTL netlist to a selected technology
4 architecture.

1 20. A machine readable medium as in claim 19 wherein said mapping is performed
2 after said partitioning.

1 21. A machine readable medium as in claim 19, wherein said method further
2 comprises:
3 performing a place and route operation after said mapping to implement said
4 ICs in said selected technology architecture.

1 22. A machine readable medium as in claim 19, wherein said method further
2 comprises:
3 optimizing a design of each of said ICs after said partitioning.

1 23. A machine readable medium as in claim 22 wherein said optimizing optimizes
2 each of said ICs by removing duplicative logic or input/outputs.

1 24. A machine readable medium as in claim 19 wherein said HDL code is created
2 without regard to said partitioning.

1 25. A machine readable medium as in claim 22 wherein said optimizing and said
2 mapping are performed after said partitioning.

1 26. A machine readable medium as in claim 19, wherein said method further
2 comprises:
3 mapping portions of said technology independent RTL netlist to a selected
4 technology architecture wherein estimates of IC resources are obtained
5 from said mapping portions and wherein said mapping portions is
6 performed after said compiling and before said mapping.

1 27. A machine readable medium as in claim 19, wherein said method further
2 comprises:

3 optimizing interconnects between modules of said technology independent
4 RTL netlist before said partitioning.

1 28. A machine readable medium as in claim 26 wherein said estimates are used to
2 decide how to perform said partitioning.

1 29. A machine readable medium as in claim 28 wherein a user considers said
2 estimates and selects a command to decide how to perform said partitioning.

1 30. A machine readable medium as in claim 19 wherein said ICs each comprise a
2 programmable logic device and wherein said method further comprises:
3 testing a prototype of a system with said ICs;
4 performing a synthesis of said HDL code to generate at least one Application
5 Specific Integrated Circuit (ASIC).

1 31. A system of designing a plurality of integrated circuits (ICs), said system
2 comprising:
3 means for displaying a representation of said plurality of ICs;
4 means for partitioning a technology independent RTL (register transfer level)
5 netlist between said plurality of ICs.

1 32. A system as in claim 31 further comprising:

2 means for compiling a hardware description language (HDL) code, wherein
3 said technology independent RTL netlist is produced after compiling
4 said HDL code.

1 33. A system as in claim 32 wherein said ICs each comprise a programmable logic
2 device.

1 34. A system as in claim 32 further comprising:
2 means for mapping said technology independent RTL netlist to a selected
3 technology architecture.

1 35. A system as in claim 34 wherein said mapping is performed after said
2 partitioning.

1 36. A system as in claim 34 further comprising:
2 means for performing a place and route operation after said mapping to
3 implement said ICs in said selected technology architecture.

1 37. A system as in claim 34 further comprising:
2 means for optimizing a design of each of said ICs after said partitioning.

1 38. A system as in claim 37 wherein said optimizing optimizes each of said ICs by
2 removing duplicative logic or input/outputs.

1 39. A system as in claim 34 wherein said HDL code is created without regard to
2 said partitioning.

1 40. A system as in claim 37 wherein said optimizing and said mapping are
2 performed after said partitioning.

1 41. A system as in claim 34 further comprising:
2 means for mapping portions of said technology independent RTL netlist to a
3 selected technology architecture wherein estimates of IC resources are
4 obtained from said mapping portions and wherein said mapping
5 portions is performed after said compiling and before said mapping.

1 42. A system as in claim 34 further comprising:
2 means for optimizing interconnects between modules of said technology
3 independent RTL netlist before said partitioning.

1 43. A system as in claim 41 wherein said estimates are used to decide how to
2 perform said partitioning.

1 44. A system as in claim 43 wherein a user considers said estimates and selects a
2 command to decide how to perform said partitioning.

1 45. A system as in claim 34 wherein said ICs each comprise a programmable logic
2 device and wherein said system further comprises:

3 means for testing a prototype of a system with said ICs;

4 means for performing a synthesis of said HDL code to generate at least one

5 Application Specific Integrated Circuit (ASIC).

1 46. A digital processing system for use in designing a plurality of integrated
2 circuits (ICs), said digital processing system comprising:

3 a display device;

4 a memory;

5 a processor coupled to said memory and to said display device, said processor

6 partitioning a technology independent RTL (register transfer level)

7 netlist between representations of said plurality of ICs, said technology

8 independent RTL netlist being stored in said memory.

1 47. A digital processing system as in claim 46 wherein said processor compiles a
2 hardware description language (HDL) code to produce said technology independent
3 RTL netlist.

1 48. A digital processing system as in claim 47 wherein said ICs each comprise a
2 programmable logic device.

1 49. A digital processing system as in claim 47 wherein said processor maps said
2 technology independent RTL netlist to a selected technology architecture.

1 50. A digital processing system as in claim 49 wherein said processor maps said
2 technology independent RTL netlist after said processor partitions said technology
3 independent RTL netlist.

1 51. A digital processing system as in claim 49 wherein said processor performs a
2 place and route operation after said processor maps said technology independent RTL
3 netlist, wherein said place and route operation creates a representation of circuitry in
4 said selected technology architecture.

1 52. A digital processing system as in claim 50 wherein said processor optimizes a
2 design of each of said ICs after said processor partitions said technology independent
3 RTL netlist.

1 53. A digital processing system as in claim 51 wherein said processor maps said
2 technology independent RTL netlist after said processor partitions said technology
3 independent RTL netlist.

1 54. A digital processing system as in claim 53 wherein said processor maps
2 portions of said technology independent RTL netlist to said selected technology

3 architecture to generate estimates of IC resources and wherein said processor maps
4 said portions after said processor compiles said HDL code.

1 55. A digital processing system as in claim 53 wherein said processor displays
2 said estimates on said display device and stores said estimates in said memory.

1 56. A digital processing system as in claim 55 wherein said processor displays
2 graphical representations of said plurality of ICs on said display device and displays
3 on said display device representations of portions of said technology independent RTL
4 netlist and wherein said processor performs said partitioning in response to a
5 command from a user.

1 57. A method of designing a plurality of integrated circuits (ICs), said method
2 comprising:
3 compiling a hardware description language (HDL) code to produce an RTL
4 netlist representation which specifies said plurality of ICs;
5 selecting logic designed for placement on one of said plurality of ICs and
6 replicating said logic for placement on another one of said plurality of
7 ICs.

1 58. A method as in claim 57 further comprising:
2 partitioning said RTL netlist representation to specify said plurality of ICs.

1 59. A method as in claim 58 further comprising:
2 mapping said RTL netlist representation to a selected technology architecture,
3 wherein said RTL netlist representation is technology independent.

1 60. A method as in claim 59 wherein each of said plurality of ICs comprises a
2 programmable logic device.

1 61. A method as in claim 59 wherein said mapping is performed after said
2 partitioning.

1 62. A method as in claim 61 further comprising:
2 performing a place and route operation after said mapping to implement logic
3 in said plurality of ICs.

1 63. A method as in claim 62 further comprising:
2 optimizing a design of each of said plurality of ICs after said partitioning.

1 64. A method as in claim 57 wherein said HDL code is created without specifying
2 a plurality of ICs.

1 65. A method as in claim 59 further comprising:
2 mapping portions of said RTL netlist representation to said selected technology
3 architecture wherein estimates of IC resources are obtained from said

4 mapping portions and wherein said mapping portions is performed
5 after said compiling and before said mapping.

1 66. A method as in claim 65 wherein said estimates are used to decide how to
2 perform said selecting and replicating.

1 67. A method as in claim 60 further comprising:
2 testing a prototype of a system with said plurality of ICs;
3 performing, after said testing, a synthesis on said HDL code to generate at
4 least one Application Specific Integrated Circuit (ASIC).

1 68. A machine readable medium containing a plurality of executable instructions,
2 which when executed on a digital processing system cause said digital processing
3 system to perform a method of designing a plurality of integrated circuits (ICs), said
4 method comprising:
5 compiling a hardware description language (HDL) code to produce an RTL
6 netlist representation which specifies said plurality of ICs;
7 selecting logic designed for placement on one of said plurality of ICs and
8 replicating said logic for placement on another one of said plurality of
9 ICs.

1 69. A machine readable medium as in claim 68, said method further comprising:
2 partitioning said RTL netlist representation to specify said plurality of ICs.

1 70. A machine readable medium as in claim 69, said method further comprising:
2 mapping said RTL netlist representation to a selected technology architecture,
3 wherein said RTL netlist representation is technology independent.

1 71. A machine readable medium as in claim 70, wherein each of said plurality of
2 ICs comprises a programmable logic device.

1 72. A machine readable medium as in claim 70, wherein said mapping is
2 performed after said partitioning.

1 73. A machine readable medium as in claim 72, said method further comprising:
2 performing a place and route operation after said mapping to implement logic
3 in said plurality of ICs.

1 74. A machine readable medium as in claim 73, said method further comprising:
2 optimizing a design of each of said plurality of ICs after said partitioning.

1 75. A machine readable medium as in claim 68 wherein said HDL code is created
2 without specifying a plurality of ICs.

1 76. A machine readable medium as in claim 70, said method further comprising:

2 mapping portions of said RTL netlist representation to said selected technology
3 architecture wherein estimates of IC resources are obtained from said
4 mapping portions and wherein said mapping portions is performed
5 after said compiling and before said mapping.

1 77. A machine readable medium as in claim 76 wherein said estimates are used to
2 decide how to perform said selecting and replicating.

1 78. A machine readable medium as in claim 71, said method further comprising:
2 testing a prototype of a system with said plurality of ICs;
3 performing, after said testing, a synthesis on said HDL code to generate at
4 least one Application Specific Integrated Circuit (ASIC).

1 79. A system for designing a plurality of integrated circuits (ICs), said system
2 comprising:
3 means for compiling a hardware description language (HDL) code to produce
4 an RTL netlist representation which specifies said plurality of ICs;
5 means for selecting logic designed for placement on one of said plurality of
6 ICs and replicating said logic for placement on another one of said
7 plurality of ICs.

1 80. A system as in claim 79 further comprising:

2 means for partitioning said RTL netlist representation to specify said plurality
3 of ICs.

1 81. A system as in claim 80 further comprising:
2 means for mapping said RTL netlist representation to a selected technology
3 architecture, wherein said RTL netlist representation is technology
4 independent.

1 82. A system as in claim 81 wherein each of said plurality of ICs comprises a
2 programmable logic device.

1 83. A system as in claim 81 wherein said mapping is performed after said
2 partitioning.

1 84. A system as in claim 83 further comprising:
2 means for performing a place and route operation after said mapping to
3 implement logic in said plurality of ICs.

1 85. A system as in claim 84 further comprising:
2 means for optimizing a design of each of said plurality of ICs after said
3 partitioning.

1 86. A system as in claim 79 wherein said HDL code is created without specifying
2 a plurality of ICs.

1 87. A system as in claim 81 further comprising:
2 means for mapping portions of said RTL netlist representation to said selected
3 technology architecture wherein estimates of IC resources are obtained
4 from said mapping portions and wherein said mapping portions is
5 performed after said compiling and before said mapping.

1 88. A system as in claim 87 wherein said estimates are used to decide how to
2 perform said selecting and replicating.

1 89. A system as in claim 82 further comprising:
2 means for testing a prototype of a system with said plurality of ICs;
3 means for performing, after said testing, a synthesis on said HDL code to
4 generate at least one Application Specific Integrated Circuit (ASIC).

1 90. A method of designing a plurality of integrated circuits (ICs), said method
2 comprising:
3 compiling a hardware description language (HDL) code to produce an RTL
4 netlist representation;
5 selecting one RTL component in said RTL netlist representation and splitting
6 said one RTL component into a first RTL component designed for

7 placement on a first IC and a second RTL component designed for
8 placement on a second IC.

1 91. A method as in claim 90 further comprising:
2 partitioning said RTL netlist representation to specify said first IC and said
3 second IC.

1 92. A method as in claim 91 further comprising:
2 mapping said RTL netlist representation to a selected technology architecture,
3 wherein said RTL netlist representation is technology independent.

1 93. A method as in claim 92 wherein each of said first IC and said second IC
2 comprises a programmable logic device.

1 94. A method as in claim 92 wherein said mapping is performed after said
2 partitioning.

1 95. A method as in claim 94 further comprising:
2 performing a place and route operation after said mapping to implement logic
3 for said first RTL component for placement on said first IC and to
4 implement logic for said second RTL component for placement on said
5 second IC.

1 96. A method as in claim 92 wherein said one RTL component is created as a
2 unitary object after said compiling.

1 97. A method as in claim 92 wherein said one RTL component is a large RTL
2 component having more inputs/outputs than is available on said first IC or said second
3 IC.

1 98. A method as in claim 92 further comprising:
2 mapping portions of said RTL netlist representation to said selected technology
3 architecture wherein estimates of IC resources are obtained from said
4 mapping portions and wherein said mapping portions is performed
5 after said compiling and before said mapping.

1 99. A method as in claim 98 wherein said selecting and said splitting are
2 performed after said compiling and before said mapping.

1 100. A method as in claim 99 wherein said estimates are used to determine how to
2 split said one RTL component.

1 101. A machine readable medium containing a plurality of executable instructions,
2 which when executed on a digital processing system cause said digital processing
3 system to perform a method of designing a plurality of integrated circuits (ICs), said
4 method comprising:

5 compiling a hardware description language (HDL) code to produce an RTL
6 netlist representation;
7 selecting one RTL component in said RTL netlist representation and splitting
8 said one RTL component into a first RTL component designed for
9 placement on a first IC and a second RTL component designed for
10 placement on a second IC.

1 102. A machine readable medium as in claim 101, wherein said method further
2 comprises:

3 partitioning said RTL netlist representation to specify said first IC and said
4 second IC.

1 103. A machine readable medium as in claim 102, wherein said method further
2 comprises:

3 mapping said RTL netlist representation to a selected technology architecture,
4 wherein said RTL netlist representation is technology independent.

1 104. A machine readable medium as in claim 103 wherein each of said first IC and
2 said second IC comprises a programmable logic device.

1 105. A machine readable medium as in claim 103 wherein said mapping is
2 performed after said partitioning.

1 106. A machine readable medium as in claim 105, wherein said method further
2 comprises:
3 performing a place and route operation after said mapping to implement logic
4 for said first RTL component for placement on said first IC and to
5 implement logic for said second RTL component for placement on said
6 second IC.

1 107. A machine readable medium as in claim 103 wherein said one RTL component
2 is created as a unitary object after said compiling.

1 108. A machine readable medium as in claim 103 wherein said one RTL component
2 is a large RTL component having more inputs/outputs than is available on said first IC
3 or said second IC.

1 109. A machine readable medium as in claim 103, wherein said method further
2 comprises:
3 mapping portions of said RTL netlist representation to said selected technology
4 architecture wherein estimates of IC resources are obtained from said
5 mapping portions and wherein said mapping portions is performed
6 after said compiling and before said mapping.

1 110. A machine readable medium as in claim 109 wherein said selecting and said
2 splitting are performed after said compiling and before said mapping.

1 111. A machine readable medium as in claim 110 wherein said estimates are used to
2 determine how to split said one RTL component.

1 112. A system for designing a plurality of integrated circuits (ICs), said system
2 comprising:
3 means for compiling a hardware description language (HDL) code to produce
4 an RTL netlist representation;
5 means for selecting one RTL component in said RTL netlist representation and
6 splitting said one RTL component into a first RTL component designed
7 for placement on a first IC and a second RTL component designed for
8 placement on a second IC.

1 113. A system as in claim 112 further comprising:
2 means for partitioning said RTL netlist representation to specify said first IC
3 and said second IC.

1 114. A system as in claim 113 further comprising:
2 means for mapping said RTL netlist representation to a selected technology
3 architecture, wherein said RTL netlist representation is technology
4 independent.

1 115. A system as in claim 114 wherein each of said first IC and said second IC
2 comprises a programmable logic device.

1 116. A system as in claim 114 wherein said mapping is performed after said
2 partitioning.

1 117. A system as in claim 116 further comprising:
2 means for performing a place and route operation after said mapping to
3 implement logic for said first RTL component for placement on said
4 first IC and to implement logic for said second RTL component for
5 placement on said second IC.

1 118. A system as in claim 114 wherein said one RTL component is created as a
2 unitary object after said compiling.

1 119. A system as in claim 114 wherein said one RTL component is a large RTL
2 component having more inputs/outputs than is available on said first IC or said second
3 IC.

1 120. A system as in claim 114 further comprising:
2 means for mapping portions of said RTL netlist representation to said selected
3 technology architecture wherein estimates of IC resources are obtained

4 from said mapping portions and wherein said mapping portions is
5 performed after said compiling and before said mapping.

1 121. A system as in claim 120 wherein said selecting and said splitting are
2 performed after said compiling and before said mapping.

1 122. A system as in claim 121 wherein said estimates are used to determine how to
2 split said one RTL component.

1 123. A method of designing a plurality of integrated circuits (ICs), said method
2 comprising:
3 assigning a first portion of a technology independent RTL (register transfer
4 level) netlist for placement on a first IC of said plurality of ICs;
5 assigning a second portion of said technology independent RTL netlist to a
6 second IC of said plurality of ICs.

1 124. A method as in claim 123 further comprising:
2 compiling a hardware description language (HDL) code, wherein said
3 technology independent RTL netlist is produced after compiling said
4 HDL code.

1 125. A method as in claim 124 wherein said ICs each comprise a programmable
2 logic device.

- 1 126. A method as in claim 124 further comprising:
2 mapping said technology independent RTL netlist to a selected technology
3 architecture.
- 1 127. A method as in claim 126 wherein said mapping is performed after said
2 assigning.
- 1 128. A method as in claim 126 further comprising:
2 performing a place and route operation after said mapping to implement said
3 ICs in said selected technology architecture.
- 1 129. A method as in claim 126 further comprising:
2 optimizing a design of each of said ICs after said assigning.
- 1 130. A method as in claim 129 wherein said optimizing optimizes each of said ICs
2 by removing duplicative logic or input/outputs.
- 1 131. A method as in claim 126 wherein said HDL code is created without regard to
2 said assigning.
- 1 132. A method as in claim 129 wherein said optimizing and said mapping are
2 performed after said assigning.

1 133. A method as in claim 126 further comprising:
2 mapping portions of said technology independent RTL netlist to a selected
3 technology architecture wherein estimates of IC resources are obtained
4 from said mapping portions and wherein said mapping portions is
5 performed after said compiling and before said mapping.

1 134. A method as in claim 126 further comprising:
2 optimizing interconnects between modules of said technology independent
3 RTL netlist before said assigning.

1 135. A method as in claim 133 wherein said estimates are used to decide how to
2 perform said assigning.

1 136. A method as in claim 135 wherein a user considers said estimates and selects a
2 command to decide how to perform said assigning.

1 137. A method as in claim 126 wherein said ICs each comprise a programmable
2 logic device and wherein said method further comprises:
3 testing a prototype of a system with said ICs;
4 performing a synthesis of said HDL code to generate at least one Application
5 Specific Integrated Circuit (ASIC).